Comparative Performance Analysis of Active Learning Strategies for the Entity Recognition Task

Philipp Kohl[1][a], Yoka Krämer[1][b], Claudia Fohry[2] and Bodo Kraft[1]
[1] *FH Aachen, University of Applied Sciences, 52428 Jülich, Germany*
[2] *University of Kassel, 34121 Kassel, Germany*
{*p.kohl, y.kraemer, kraft*}*@fh-aachen.de, fohry@uni-kassel.de*

Keywords: Active Learning, Selective Sampling, Named Entity Recognition, Span Labeling, Annotation Effort.

Abstract: Supervised learning requires a lot of annotated data, which makes the annotation process time-consuming and expensive. Active Learning (AL) offers a promising solution by reducing the number of labeled data needed while maintaining model performance. This work focuses on the application of supervised learning and AL for (named) entity recognition, which is a subdiscipline of Natural Language Processing (NLP). Despite the potential of AL in this area, there is still a limited understanding of the performance of different approaches. We address this gap by conducting a comparative performance analysis with diverse, carefully selected corpora and AL strategies. Thereby, we establish a standardized evaluation setting to ensure reproducibility and consistency across experiments. With our analysis, we discover scenarios where AL provides performance improvements and others where its benefits are limited. In particular, we find that strategies including historical information from the learning process and maximizing entity information yield the most significant improvements. Our findings can guide researchers and practitioners in optimizing their annotation efforts.

1 INTRODUCTION

Supervised model training is a widely adopted approach that requires annotated data. This data is obtained through an annotation process, which often necessitates the expertise of domain specialists, particularly in fields such as biology, medicine, and law. The involvement of experts is costly (Finlayson and Erjavec, 2017). To alleviate the costs, researchers have introduced various methods to reduce the annotation effort (Sintayehu and Lehal, 2021; Lison et al., 2021; Feng et al., 2021; Wang et al., 2019; Yang, 2021). A popular method is Active Learning (AL). It is based on the principle that not all data points are equally valuable for the learning process and thus strives to select a particularly informative subset for annotation (Settles, 2009).

Despite the development of numerous AL strategies, their performance across different use cases is not well understood. We consider the case of *entity recognition (ER)* in NLP and conduct a comparative performance analysis (Jehangir et al., 2023). A representative subset of corpora and AL strategies is included, which we selected from a specialized scoping review (Kohl et al., 2024).

Our contributions are as follows:

- We establish a comprehensive framework for evaluating AL strategies for ER. This includes identifying a subset of datasets (*corpora*) that covers a wide range of domains (e.g., newspapers, medicine, etc.) and significant AL parameters, selecting a broad range of AL strategies for diverse evaluation, and designing a suitable model architecture that balances both performance and runtime for testing.
- We conduct an extensive analysis to determine the best-performing AL strategies for ER, identifying strategies that perform consistently well across different domains. We also evaluate the robustness and stability of these strategies, considering the impact of random processes in model training and the AL process.

The paper is structured as follows: Section 2 starts with an overview of the research field and related work. Then, in Section 3, we delve into the fundamental concepts of AL, ER, and the *Active Learning Evaluation (ALE) Framework*. Afterward, Section 4 explains how we selected the subset of corpora

[a] https://orcid.org/0000-0002-5972-8413
[b] https://orcid.org/0009-0006-7326-3268

Kohl, P., Krämer, Y., Fohry, C. and Kraft, B.
Comparative Performance Analysis of Active Learning Strategies for the Entity Recognition Task.
DOI: 10.5220/0013068200003838

In *Proceedings of the 16th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2024) - Volume 1: KDIR*, pages 480-488
ISBN: 978-989-758-716-0; ISSN: 2184-3228

and strategies tested in this study. We then present a description of the experimental setup in Section 5. While Section 6 presents the results and analyzes our experimental findings, Section 7 concludes the paper.

Our results, including code, figures, and extensive tables, can be found on GitHub[1].

2 RELATED WORK

Researchers introduced numerous AL strategies for areas such as computer vision or NLP (Settles, 2009; Ren et al., 2022; Schröder and Niekler, 2020; Zhang et al., 2022; Kohl et al., 2024). The strategies have been classified into taxonomies to provide a structured domain understanding. However, the existing surveys typically refrain from ranking the strategies based on their efficacy (Zhan et al., 2022). There is a general lack of comparative performance data. While any new strategy is backed by performance data, these typically refer to a limited and arbitrary subset of existing strategies. Direct comparisons are further complicated by variability in parameter selection and implementation details. The present paper helps to close this gap by providing a systematically designed comparative performance analysis for AL strategies in the ER domain. The limited knowledge of the relative performance of advanced AL methods may explain why current annotation tools such as Inception (Klie et al., 2018), Prodigy (Montani and Honnibal,), and Doccano (Nakayama et al., 2018) focus on basic AL strategies, potentially overlooking more sophisticated ones.

Several frameworks support the implementation and evaluation of AL strategies in other areas. *libact* (Yang et al., 2017) focuses on comparing AL strategies with scikit-learn models, while DeepAL (Huang, 2021; Zhan et al., 2022) is tailored for image vision tasks. We utilize the Active Learning Evaluation (ALE) framework (Kohl et al., 2023), which has a sophisticated, modular design, supports integration with various deep learning libraries and cloud computing environments, and has a strong focus on reproducible research.

Besides AL, there are other approaches that can reduce the annotation effort: semi-supervised learning (Sintayehu and Lehal, 2021) leverages a small labeled dataset to annotate unlabeled data, and weak supervision (Lison et al., 2021) uses heuristics or labeling functions to annotate data automatically. Data augmentation (Feng et al., 2021) generates new examples by replacing words or reformulating sentences, enhancing the training dataset without additional manual effort. Zero-shot (Wang et al., 2019) and few-shot learning (Yang, 2021; Brown et al., 2020) techniques transfer knowledge from one domain to another, reducing the need for extensive new datasets. Large language models (LLMs) are inherently few-shot learners (Brown et al., 2020), but they are not always applicable due to offline scenarios, hardware limitations, or the need for smaller models in specialized domains (Jayakumar et al., 2023).

3 FUNDAMENTALS

In this section, we introduce core concepts and a common taxonomy of AL, which will be used in Section 4. We also define and embed the ER task into the active learning domain. Finally, we provide some details on the ALE framework.

3.1 Active Learning

Active learning (AL) addresses the reduction of annotation effort and, therefore, is embedded into the *annotation process* (Settles, 2009). This process consists of three steps: (a) selecting unlabeled documents (*batch*), (b) annotating these documents, and (c) training a classifier. These steps are repeated until performance metrics (e.g., F1 score) reach a desired value. AL modifies step (a) so that data points are selected with an AL strategy instead of randomly or sequentially. AL is based on the assumption that different data points have different information gains for the learning process. The AL strategies quantify these gains (Settles, 2009; Finlayson and Erjavec, 2017).

The AL strategies can be divided into three categories (Settles, 2009; Zhan et al., 2022; Kohl et al., 2024):

Exploitation depends on model feedback (e.g., confidence scores) to compute an informativeness score. For example, *least confidence* selects data points the model is most uncertain about.

Exploration is solely based on the corpora and uses similarities and dissimilarities between data points. For example, some strategies embed the data points into a high-dimensional vector space and utilize cluster methods to select a batch of data points from different clusters.

Hybrid strategies combine exploitation and exploration approaches, for instance, by merging their scores. Several hybrid approaches start with exploration to identify a subset of the data points, which is then analyzed using exploitation. This way the need

[1] https://github.com/philipp-kohl/comparative-performance-analysis-al-ner

for costly model feedback is reduced to the selected subset.

3.2 Entity Recognition

Entity Recognition (ER) is a subtask of *information extraction* (Jehangir et al., 2023). Given some unstructured text, ER finds arbitrary, predefined domain-specific *entities* (e.g., persons, diseases, time units, etc.). On the technical level (see Figure 1), a model tokenizes the text and classifies these tokens. Thus, the model feedback (e.g., confidence scores) is present for each token.

AL strategies select whole documents for annotation. Some AL strategies rely on model feedback, which requires to aggreate the token-wise information to a document-wise score. Figure 1 visualizes the aggregation process.

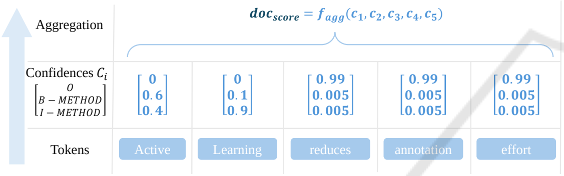

Aggregation
$doc_{score} = f_{agg}(c_1, c_2, c_3, c_4, c_5)$
Confidences C_i
$\begin{bmatrix} O \\ B-METHOD \\ I-METHOD \end{bmatrix}$
$\begin{bmatrix} 0 \\ 0.6 \\ 0.4 \end{bmatrix}$
$\begin{bmatrix} 0 \\ 0.1 \\ 0.9 \end{bmatrix}$
$\begin{bmatrix} 0.99 \\ 0.005 \\ 0.005 \end{bmatrix}$
$\begin{bmatrix} 0.99 \\ 0.005 \\ 0.005 \end{bmatrix}$
$\begin{bmatrix} 0.99 \\ 0.005 \\ 0.005 \end{bmatrix}$
Tokens
Active
Learning
reduces
annotation
effort

Figure 1: Tokenized text on the lowest level (whitespace tokenization for simplicity) on which the model infers predictions with the IOB2 (Ramshaw and Marcus, 1995) schema and computes confidence scores. At the top level, an aggregation function would compute a document-wise score based on the confidences per token.

3.3 Active Learning Evaluation Framework

We use the Active Learning Evaluation (ALE) framework (Kohl et al., 2023) for comparing different AL strategies against each other. ALE simulates the annotation process (see Subsection 3.1), which we call an *AL cycle*: (a) proposing new data points using an AL strategy. (b) annotating the data. Instead of forwarding the selected batches to human annotators, ALE uses provided gold labels of the corpora for the simulation. (c) Training and evaluation of the model.

Figure 2 gives an overview of ALE. The framework spans different stages. The first stage represents an experiment, which simulates a single strategy. The experiment follows a pipeline approach to preprocess the data and start so-called *seed runs*. Each seed run simulates one annotation process (*AL cycle*) with some random seed. Multiple seed runs are conducted to assess the stability and robustness of the AL strategies. Table 1 shows the connection between seed runs and AL cycles: A row represents the annotation process for a single seed with a growing corpus, while the column provides information on the robustness.

Table 1: Example F1 scores for seed runs across AL cycle iterations in a single experiment. Each cell shows the F1 score measured on the test corpus after each data proposal. For instance, AL cycle 2 represents the F1 scores after the second data proposal.

Seed Run	**AL Cycle 1**	**AL Cycle 2**	...	**ALCycle N**
Seed 1	0.01	0.05	...	0.85
Seed 2	0.01	0.06	...	0.83
...	...	...	...	...
Seed M	0.02	0.04	...	0.87

ALE has many configuration parameters. These and the corresponding experimental outcomes are reported to MLflow[2], which is an MLOps platform that supports reproducible research. The two core parameters are the *seeds* and the *step size*. The seeds-parameter is an integer list defining which and how many seed runs ALE starts. The step size defines how many documents the AL strategy selects in step (a) of the AL cycle.

ALE comes with an implementation for *spaCy*[3], which we have replaced by *PyTorch Lightning*[4] as deep learning library for step (c) of the AL cycle. PyTorch Lightning gives us finer control of the learning process.

The framework provides evaluation functions to address two critical aspects of AL: data bias and model calibration. It is crucial to avoid AL strategies that exacerbate existing biases within the dataset (see Section 6). Additionally, reliable model feedback requires well-calibrated models. To assess model calibration, ALE employs the *expected calibration error* (ECE) and *reliability diagrams* (Wang et al., 2021).

4 SELECTION OF CORPORA & STRATEGIES

We base our selection of corpora and strategies on the scoping review (Kohl et al., 2024), which reviewed 62 papers and collected information about the used AL strategies and other aspects of the evaluation environment.

4.1 Corpora

(Kohl et al., 2024) provide a collection of 26 publicly available corpora used to evaluate AL strategies for

[2] https://mlflow.org/

[3] https://spacy.io/

[4] https://lightning.ai/docs/pytorch/stable/

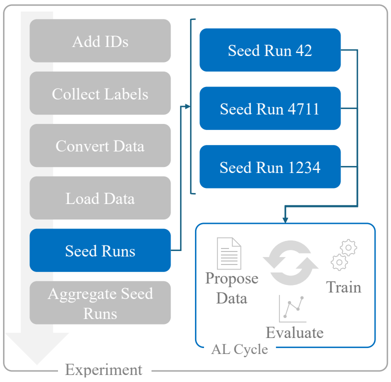


Figure 2: The ALE framework introduces three key concepts: *Experiments*, *Seed Runs*, and *AL cycles*. Each experiment involves a pipeline execution, with Seed Runs as the core element. A single Seed Run represents one AL cycle.

ER. We selected seven corpora based on the following criteria: frequency of use, diversity of domains (e.g., newspapers, medicine, social media), varying language complexity measured by the *moving average type-token ratio (MATTR)* (Covington and McFall, 2010; Kettunen, 2014)), label complexity and distribution (e.g., number of labels per sample), and average document length (limited to 512 tokens for compatibility with our model). The selected corpora are CoNLL2003, MedMentions, JNLPBA, GermEval, SCIERC, WNUT, and AURC-7. Further details are provided in Table 2.

4.2 Strategies

For strategy selection, we followed (Kohl et al., 2024), which highlights a focus on uncertainty exploitation strategies, particularly *entropy*, *margin*, and *least confidence*. These strategies use token-level confidences to compute scores, which are aggregated using methods such as average, minimum, maximum, sum, and standard deviation (Subsection 3.2). In addition to these three uncertainty strategies, we included count-based, round-robin, and two specialized strategies considering past predictions, as well as three exploration and two hybrid approaches.

Exploitation Strategies.

Least Confidence (LC) measures the uncertainty of the model for each token. The strategy strives to select documents the model is most uncertain about to receive a high information gain (Esuli et al., 2010; Şapci et al., 2023).

Margin Confidence computes the difference (margin) of the confidences for the two most probable labels per token. The intention is that a confident decision would have a high margin (e.g., $0.93 - 0.03 = 0.9$) because the decision boundary is learned well, while not confident decisions have very low margins (e.g., $0.45 - 0.4 = 0.05$). The strategy selects documents with low aggregated margins (Settles, 2009; Şapci et al., 2023).

Entropy Confidence uses the Shannon entropy to quantify the expected information gain. The strategy selects documents with a high entropy (Yao et al., 2020; Şapci et al., 2023).

Max Tag Count sums the number of entities the model predicts in a document (label different from the O-tag). The strategy favors documents with many entities because the authors hypothesize that the information gain is higher (Esuli et al., 2010).

Round Robin by Label strives to achieve a balanced distribution of labels in the batches. The strategy employs a round-robin approach to select documents based on their labels. The strategy maintains a score for each label per document. This differs from the previous strategies, which compute a single score per document (Esuli et al., 2010).

Fluctuation of Historical Sequence measures the uncertainty over the last n predictions (*historical*) instead of only considering the current prediction. The authors define a formula for a weighted sum of the current confidence and the historical confidence scores. The intuition is that volatile confidence scores indicate a higher impact on the learning process than stable ones because they might influence the decision boundary (Yao et al., 2020).

Tag Flip of Historical Sequence measures the instability of the model's decisions for a document. It counts the label changes (*tag flip*) for each token in a document across the last n predictions. Documents with many flips can be an indicator to influence the decision boundaries and, therefore, are beneficial for the training process (Zheng et al., 2018).

Exploration Strategies.

Diversity embeds the dataset into a vector space and precomputes pair-wise cosine similarities. The strategy selects data points that are most dissimilar to already labeled data points. In that way, the dataset should be diverse (Chen et al., 2015).

Maximum Representativeness-Diversity extends the previous strategy by adding the condition to not only select data points that are most dissimilar to already labeled data points (diversity) but also most similar to unlabeled documents (representative). The authors (Kholghi et al., 2015) use the product of the diversity and the representative score as document score.

K-Means Cluster Centroids embeds the data points into a vector space and clusters them with the k-means algorithm. The strategy selects data points nearest to cluster centroids (Van Nguyen et al., 2022).

Table 2: Overview of the seven selected corpora: Besides the domain as a selection criterion, the characteristics highlighted in bold also served as criteria. The row *number of labels* also states information about the label balance.

Corpus	**CoNLL03**	**MedMent.**	**JNLPBA**	**SCIERC**	**WNUT16**	**GermEval**	**AURC7**
Domain	**News**	**Medicine**	**Bio-medicine**	Scientific papers	Twitter posts	Encyclo-pedia	Politics
MATTR	**0.96**	**0.77**	0.9	0.79	0.95	0.96	0.89
Size (s=sample, t=token)	20744 s ⌀ **15 t**	4392 s ⌀ **275 t**	**22402 s** ⌀ 26 t	500 s ⌀ **131 t**	7244 s ⌀ 18 t	31300 s ⌀ 19 t	7977 s ⌀ 27 t
# of labels	4 (unbal.)	1 (bal.)	5 (unbal.)	6 (unbal.)	**10** (unbal.)	3 (unbal.)	2 **(bal.)**
Language	English	English	English	English	English	**German**	English
Data ratio without labels	0.205	**0**	0.113	0.002	0.537	0.411	0.436
# of labels per sample	1.691	**80**	2.674	16.188	0.771	1.206	0.634

Hybrid Strategies.

Representative LC sequentially applies an exploration and then an exploitation strategy. At first, the exploration strategy selects data points that represent the unlabeled documents best. The least confidence strategy selects data points from this subset the model is most uncertain about (Kholghi et al., 2017).

Information Density uses a combination of the representative and the entropy strategy. For each document, the strategy independently computes the cosine similarity with the unlabeled dataset and the entropy score. Afterward, the product of these scores represents the document (Settles and Craven, 2008).

5 EXPERIMENTAL SETUP

We conducted four experiment series, which are illustrated in Figure 3: The results of the first three *pre-series* led to our *standard series*, which we applied to all strategies. For all experiment series, we defined two test concepts:

Performance Tests: measure the F1 macro score at each iteration of the AL cycle. Following each data proposal, ALE retrains the model on the growing training corpus and evaluates the model on the corresponding complete and immutable test corpus. The scores are averaged across the seed runs (Table 1). Good-performing AL strategies show a steeper increase than the randomizer in model performance (see Figure 5).

Variance Tests: measure the variance and standard deviation of the F1 macro scores for each iteration of the AL cycle across the seed runs (Table 1). AL strategies with lower variance are preferable because they do not seem to be sensitive to random processes. We also call strategies fulfilling this characteristic *robust*.

The *Model Architecture* series explored various models from the RoBERTa family (Liu et al., 2019), taking into account the large number of experiments and their associated runtime. To ensure reliable confidence estimates, we tested label smoothing (Wang et al., 2021) and a CRF layer (Liu et al., 2022). Label smoothing yielded better model calibration. In the *Seed Settings* series, we assessed the number of seed runs required to obtain stable variance and performance estimates. Additionally, in *Aggregation Methods*, we evaluated different aggregations for uncertainty strategies, selecting only the most effective ones for use in the *Comprehensive Comparison*: We summarize the main parameters as follows: We use the *Distil RoBERTa Base* model(Liu et al., 2019; Sanh et al., 2020)[5] with label smoothing of 0.2. To realize a fair comparison between the different strategies, we set fixed hyperparameters for the model. Therefore, we always used 50 training epochs, a learning rate of $2e-5$, and a weight decay of 0.01 as recommended by (Liu et al., 2019; Kaddour et al., 2023). We used a batch size of 64. For ALE we use 3 seed runs for performance tests and 20 seed runs for variance tests. We chose the *step size* per corpus so that each data proposal delivers a similar amount of tokens.

At this stage, we use only the best-performing aggregation method for the uncertainty strategies found in the pre-series *Aggregation Methods*. This results in 12 strategies. For each strategy, we run 2 variance tests and 7 performance tests. For the randomizer baseline, we only conducted the performance tests. This results in 115 single experiments.

We used a workstation with 96 CPU cores and 3 Nvidia Quadro RTX 8000, each with 48GB of VRAM. The experiments took about 720 hours (30 days).

[5] https://huggingface.co/distilbert/distilroberta-base

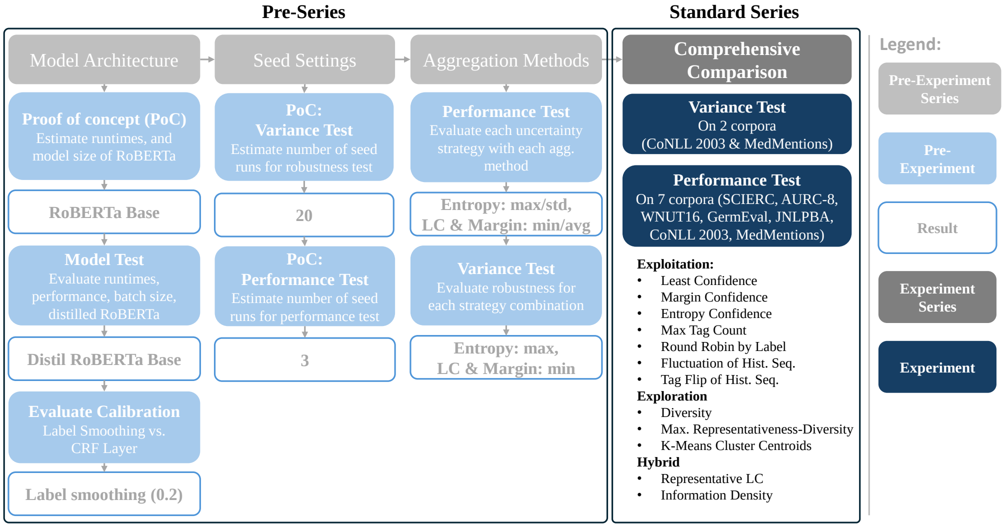

Pre-Series
Standard Series
Model Architecture
Seed Settings
Aggregation Methods
Comprehensive Comparison
Proof of concept (PoC)
Estimate runtimes, and model size of RoBERTa
PoC: Variance Test
Estimate number of seed runs for robustness test
Performance Test
Evaluate each uncertainty strategy with each agg. method
Variance Test
On 2 corpora (CoNLL 2003 & MedMentions)
RoBERTa Base
20
Entropy: max/std, LC & Margin: min/avg
Performance Test
On 7 corpora (SCIERC, AURC-8, WNUT16, GermEval, JNLPBA, CoNLL 2003, MedMentions)
Model Test
Evaluate runtimes, performance, batch size, distilled RoBERTa
PoC: Performance Test
Estimate number of seed runs for performance test
Variance Test
Evaluate robustness for each strategy combination
Distil RoBERTa Base
3
Entropy: max, LC & Margin: min
Evaluate Calibration
Label Smoothing vs. CRF Layer
Label smoothing (0.2)
Exploitation:
Least Confidence
Margin Confidence
Entropy Confidence
Max Tag Count
Round Robin by Label
Fluctuation of Hist. Seq.
Tag Flip of Hist. Seq.
Exploration
Diversity
Max. Representativeness-Diversity
K-Means Cluster Centroids
Hybrid
Representative LC
Information Density
Legend:
Pre-Experiment Series
Pre-Experiment
Result
Experiment Series
Experiment

Figure 3: Process to derive our standard evaluation setting, which was applied to each selected strategy.

6 RESULTS

The following sections describe our results regarding the performance, robustness, and data bias of the considered AL strategies.

6.1 Performance and Robustness Comparison

We assessed the performance with two methods: *Area under the learning curve (AUC)* and *Wilcoxon Signed-Rank Test (WSRT)*. AUC serves as an empirical measure to compare different strategies with each other based on the F1 macro score depending on the number of data points (see Figure 5). The larger the area under the curve, the better the strategy (Settles and Craven, 2008). The authors of (Rainio et al., 2024) recommend the WSRT to compare two models with each other based on evaluation metrics (here F1 macro score). We use it to determine which strategies are statistically significantly better than the randomizer. Then, AUC ranks these AL strategies. Figure 4 depicts the performance of each strategy and corpus. In the following, we call each combination of AL strategy and corpus a *use case* (single cell in the figure), and a *domain* is represented by a corpus and constitutes a row in the figure.

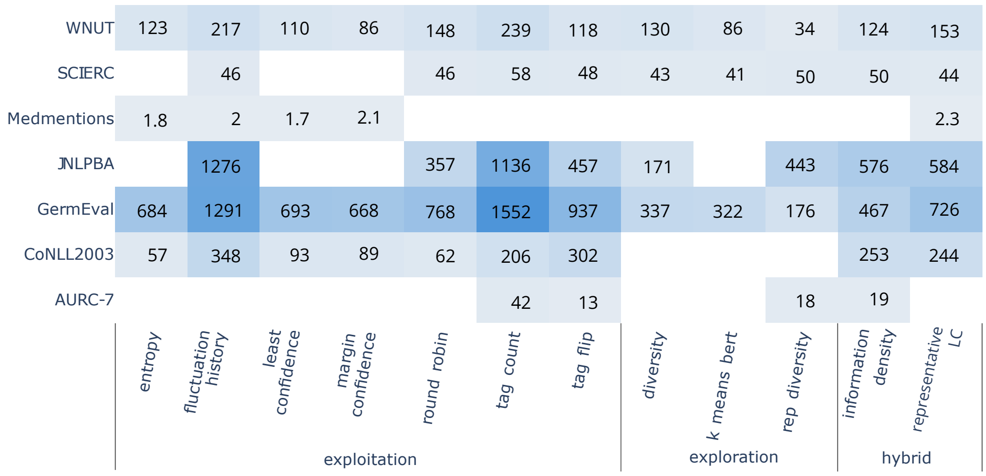
Exploration strategies show the smallest benefit. Among the selected subset of strategies — diversity (*diversity*), representative (*k means bert*), and their combination (*rep diversity*) — the combination performed best across various domains, improving 5 out of 7 use cases, while the other two improved only 3 to 4 use cases. A more extensive evaluation of further exploration strategies could provide deeper insights into this area.

The selected hybrid approaches have shown similar performance. Both improved 6 out of 7 use cases. The sequential approach (*representative LC*) was slightly better.

Among the exploitation strategies, three exhibit strong performance (*fluctuation history*, *tag count*, and *tag flip*), especially for the corpora GermEval and JNLPBA. Across the domains, they improved 6 out of 7 use cases. The other strategies show a moderate impact. Based on these results, it seems helpful to use historical information (fluctuation or flips) and documents with many entities (tag count).

We compared the hybrid strategies and their underlying exploitation methods. The integration of an exploitation approach with an exploration approach appears to extend the coverage across the use cases. For instance, the representative LC strategy, which utilizes the *least confidence* strategy, improved performance in 6 out of 7 use cases. When least confidence is applied alone, it improved 4 out of 7 use cases. A similar pattern is observed with information density, where the combination of entropy and density information demonstrates enhanced efficacy.

From the domain perspective, we made the following observations: None of the strategies is suitable for AURC-7 and Medmentions. AURC-7 is a balanced corpus with argumentation documents: each argument follows a counter-argument. Medmentions has a very high average number of entities per document (80) with only one label. In both cases, the ran-



	entropy	fluctuation history	least confidence	margin confidence	round robin	tag count	tag flip	diversity	k means bert	rep diversity	information density	representative LC
	exploitation							exploration			hybrid	
WNUT	123	217	110	86	148	239	118	130	86	34	124	153
SCIERC		46			46	58	48	43	41	50	50	44
Medmentions	1.8	2	1.7	2.1								2.3
JNLPBA		1276			357	1136	457	171		443	576	584
GermEval	684	1291	693	668	768	1552	937	337	322	176	467	726
CoNLL2003	57	348	93	89	62	206	302				253	244
AURC-7						42	13			18	19	

Figure 4: The chart displays the performance of each strategy compared to the randomizer on each corpus. White areas indicate cases where no statistically significant improvements against the randomizer were detected. All blue-shaded areas indicate statistically significant gains. The darker the shade, the better the strategy performed against the randomizer measured by AUC differences. The AUC differences are depicted in each cell.

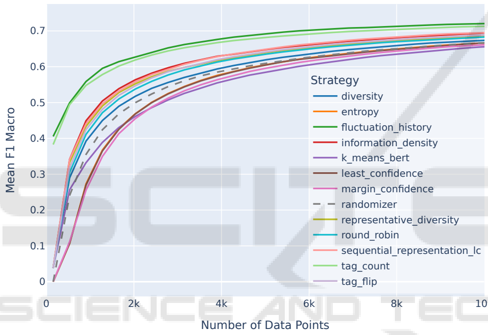


Figure 5: Mean F1 macro score on the JNLPBA test corpus. The score is averaged across three seed runs depending on the number of data points used for training (Entropy is almost fully covered by the least confidence strategy).

dom selection might gain sufficient information and cannot be improved with AL.

The strongest impact was detected for GermEval and JNLPBA, which represent the largest corpora in our test suite. See Figure 5 for the learning curves for JNLPBA as an example. Although the size of CoNLL2003 is similar to JNLPBA, we cannot see the same improvement for CoNLL2003. For GermEval and WNUT every strategy performs better than the randomizer.

We assessed the strategies' robustness via the standard deviation (see Section 5). We require that the random processes in the training process or the selection of the initial subset should not significantly impact good-performing strategies. The results show that the two best-performing strategies (fluctuation history and tag count) are also the most robust strategies. The least robust strategies are information density, representative LC, and diversity.

6.2 Bias Comparison

We also assessed the data bias and the amplification by the strategies. Inspired by (Hassan and Alikhani, 2023) on classification tasks, we extended their approach to ER. They showed that unequal label distributions infer a data bias. The authors compare the inherent label distribution of the corpora with the error distribution of the trained model. Good AL strategies should not introduce high error rates for low-frequent labels. We derived the following formula to measure the bias in our use case. Requirements:

(I) Compute the error err_l (analog to accuracy) for each label l except the O-tag. (II) Compute the normalized data distribution d_l per label l, so that you obtain values from the interval $[0,1]$ per label.

The bias per label is defined as:

$$b_l = -err_l \cdot log(d_l)$$

Errors associated with low-frequency labels tend to exacerbate bias more significantly than those linked to high-frequency labels. This measurement of bias is effective only as a comparative score within the same corpus and cannot be applied nominally across different corpora.

Our findings indicate that the strategies with the least susceptibility to bias are tag count and fluctuation history. In contrast, the strategies most amplifying bias include random selection, representative diversity, and diversity strategies. We hypothesize that the random selection strategy amplifies data bias because it mirrors the inherent data distribution. Conversely, strategies like tag count or fluctuation history appear to select beneficial subsets of data, thereby mitigating errors in low-frequency labels. This is also illustrated in Figure 5, where these strategies outperform random selection even in the region where the data sets begin to converge ($\sim$ 10k documents), further demonstrating their efficacy in reducing bias.

7 CONCLUSION

This paper conducted a comparative performance analysis of *Active Learning (AL)* strategies in the context of *entity recognition (ER)*. Based on a systematic selection of corpora and strategies, guided by a comprehensive scoping review, we conducted 115 experiments within a standardized evaluation setting. Our assessment referred to both performance and runtime. We identified conditions where AL achieved significant improvements, as well as situations where its results are more limited. Two strategies came out as clear winners: *tag count* and *fluctuation history*.

Future work may expand the evaluation to a broader range of AL strategies and corpora, including those that do not adhere to the rigorous construction standards of benchmark datasets, to explore their specific challenges.

REFERENCES

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Chen, Y., Lasko, T. A., Mei, Q., Denny, J. C., and Xu, H. (2015). A study of active learning methods for named entity recognition in clinical text. *Journal of Biomedical Informatics*, 58:11–18.

Covington, M. A. and McFall, J. D. (2010). Cutting the Gordian Knot: The Moving-Average Type–Token Ratio (MATTR). *Journal of Quantitative Linguistics*, 17(2):94–100.

Esuli, A., Marcheggiani, D., and Sebastiani, F. (2010). Sentence-based active learning strategies for information extraction. In *CEUR Workshop Proceedings*, volume 560, pages 41–45.

Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., and Hovy, E. (2021). A Survey of Data Augmentation Approaches for NLP. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Finlayson, M. A. and Erjavec, T. (2017). Overview of Annotation Creation: Processes and Tools. In Ide, N. and Pustejovsky, J., editors, *Handbook of Linguistic Annotation*, pages 167–191. Springer Netherlands, Dordrecht.

Hassan, S. and Alikhani, M. (2023). D-CALM: A Dynamic Clustering-based Active Learning Approach for Mitigating Bias. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5540–5553, Toronto, Canada. Association for Computational Linguistics.

Huang, K.-H. (2021). DeepAL: Deep Active Learning in Python.

Jayakumar, T., Farooqui, F., and Farooqui, L. (2023). Large Language Models are legal but they are not: Making the case for a powerful LegalLLM. In Preo\textcommabelowtiuc-Pietro, D., Goanta, C., Chalkidis, I., Barrett, L., Spanakis, G., and Aletras, N., editors, *Proceedings of the Natural Legal Language Processing Workshop 2023*, pages 223–229, Singapore. Association for Computational Linguistics.

Jehangir, B., Radhakrishnan, S., and Agarwal, R. (2023). A survey on Named Entity Recognition — datasets, tools, and methodologies. *Natural Language Processing Journal*, 3:100017.

Kaddour, J., Key, O., Nawrot, P., Minervini, P., and Kusner, M. J. (2023). No Train No Gain: Revisiting Efficient Training Algorithms For Transformer-based Language Models. *Advances in Neural Information Processing Systems*, 36:25793–25818.

Kettunen, K. (2014). Can Type-Token Ratio be Used to Show Morphological Complexity of Languages? *Journal of Quantitative Linguistics*, 21(3):223–245.

Kholghi, M., De Vine, L., Sitbon, L., Zuccon, G., and Nguyen, A. (2017). Clinical information extraction using small data: An active learning approach based on sequence representations and word embeddings. 68(11):2543–2556.

Kholghi, M., Sitbon, L., Zuccon, G., and Nguyen, A. (2015). External knowledge and query strategies in active learning: A study in clinical information extraction. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 143–152, New York, NY, USA. Association for Computing Machinery.

Klie, J.-C., Bugert, M., Boullosa, B., Eckart de Castilho, R., and Gurevych, I. (2018). The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico. Association for Computational Linguistics.

Kohl, P., Freyer, N., Krämer, Y., Werth, H., Wolf, S., Kraft, B., Meinecke, M., and Zündorf, A. (2023). ALE: A

Simulation-Based Active Learning Evaluation Framework for the Parameter-Driven Comparison of Query Strategies for NLP. In Conte, D., Fred, A., Gusikhin, O., and Sansone, C., editors, *Deep Learning Theory and Applications*, Communications in Computer and Information Science, pages 235–253, Cham. Springer Nature Switzerland.

Kohl, P., Krämer, Y., Fohry, C., and Kraft, B. (2024). Scoping Review of Active Learning Strategies and Their Evaluation Environments for Entity Recognition Tasks. In Fred, A., Hadjali, A., Gusikhin, O., and Sansone, C., editors, *Deep Learning Theory and Applications*, pages 84–106, Cham. Springer Nature Switzerland.

Lison, P., Barnes, J., and Hubin, A. (2021). Skweak: Weak Supervision Made Easy for NLP. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 337–346.

Liu, M., Tu, Z., Zhang, T., Su, T., Xu, X., and Wang, Z. (2022). LTP: A new active learning strategy for CRF-Based named entity recognition. *Neural Processing Letters*, 54(3):2433–2454.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Montani, I. and Honnibal, M. Prodigy: A modern and scriptable annotation tool for creating training data for machine learning models. *Prodigy*, Explosion.

Nakayama, H., Kubo, T., Kamura, J., Taniguchi, Y., and Liang, X. (2018). Doccano: Text Annotation Tool for Human. https://github.com/doccano/doccano.

Rainio, O., Teuho, J., and Klén, R. (2024). Evaluation metrics and statistical tests for machine learning. *Scientific Reports*, 14(1):6086.

Ramshaw, L. and Marcus, M. (1995). Text Chunking using Transformation-Based Learning. In *Third Workshop on Very Large Corpora*.

Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B. B., Chen, X., and Wang, X. (2022). A Survey of Deep Active Learning. *ACM Computing Surveys*, 54(9):1–40.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2020). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter.

Şapci, A., Kemik, H., Yeniterzi, R., and Tastan, O. (2023). Focusing on potential named entities during active label acquisition. *Natural Language Engineering*.

Schröder, C. and Niekler, A. (2020). A Survey of Active Learning for Text Classification using Deep Neural Networks.

Settles, B. (2009). Active Learning Literature Survey. Technical Report, University of Wisconsin-Madison Department of Computer Sciences.

Settles, B. and Craven, M. (2008). An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 1070–1079, USA. Association for Computational Linguistics.

Sintayehu, H. and Lehal, G. S. (2021). Named entity recognition: A semi-supervised learning approach. *International Journal of Information Technology*, 13(4):1659–1665.

Van Nguyen, M., Ngo, N., Min, B., and Nguyen, T. (2022). FAMIE: A Fast Active Learning Framework for Multilingual Information Extraction. In *NAACL 2022 - 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session*, pages 131–139.

Wang, D.-B., Feng, L., and Zhang, M.-L. (2021). Rethinking Calibration of Deep Neural Networks: Do Not Be Afraid of Overconfidence. In *Advances in Neural Information Processing Systems*, volume 34, pages 11809–11820. Curran Associates, Inc.

Wang, W., Zheng, V. W., Yu, H., and Miao, C. (2019). A Survey of Zero-Shot Learning: Settings, Methods, and Applications. *ACM Trans. Intell. Syst. Technol.*, 10(2):13:1–13:37.

Yang, M. (2021). A Survey on Few-Shot Learning in Natural Language Processing. In *2021 International Conference on Artificial Intelligence and Electromechanical Automation (AIEA)*, pages 294–297.

Yang, Y.-Y., Lee, S.-C., Chung, Y.-A., Wu, T.-E., Chen, S.-A., and Lin, H.-T. (2017). Libact: Pool-based Active Learning in Python.

Yao, J., Dou, Z., Nie, J., and Wen, J. (2020). Looking Back on the Past: Active Learning with Historical Evaluation Results. *IEEE Transactions on Knowledge and Data Engineering*.

Zhan, X., Wang, Q., Huang, K.-h., Xiong, H., Dou, D., and Chan, A. B. (2022). A Comparative Survey of Deep Active Learning.

Zhang, Z., Strubell, E., and Hovy, E. (2022). A Survey of Active Learning for Natural Language Processing. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6166–6190, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zheng, G., Mukherjee, S., Dong, X. L., and Li, F. (2018). OpenTag: Open attribute value extraction from product profiles. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &amp; Data Mining*, KDD '18, pages 1049–1058, New York, NY, USA. Association for Computing Machinery.